Expert Chording Text Entry on the Twiddler One–Handed Keyboard

Kent Lyons, Daniel Plaisted, Thad Starner
College of Computing and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
{kent, plaisted, thad}@cc.gatech.edu

ABSTRACT
Previously, we demonstrated that after 400 minutes of practice, ten novices averaged over 26 words per minute (wpm) for text entry on the Twiddler one–handed chording keyboard, outperforming the multi–tap mobile text entry standard. Here we present an extension of this study that examines expert chording performance. Five subjects continued the study and achieved an average rate of 47 wpm after approximately 25 hours of practice in varying conditions. One subject achieved a rate of 67 wpm, equivalent to the typing rate of the last author who has been a Twiddler user for ten years. We provide evidence that lack of visual feedback does not hinder expert typing speed and examine the potential use of multiple character chords (MCCs) to increase text entry speed. We demonstrate the effects of learning on various aspects of chording and analyze how subjects adopt a simultaneous or sequential method of pushing the individual keys during a chord.

Introduction
Wireless text messaging is becoming widespread, with predictions of a rate of over 1 trillion messages per year being reached shortly [3, 10]. However, slow text entry on mobile devices may limit the utility of upcoming services such as wireless e-mail, which is expected to be the next driver of growth in the cellular industry [1]. In this paper, we present an evaluation of a chording method of text entry on the Twiddler, a 3x4 button keypad. We present the rates of learning for chording, present data on our expert participants using multi-character chords (MCCs), and examine the effects of varying visual feedback on expert typing speeds.

Twiddler Chording
Many wearable computer users [4, 13] type with the HandyKey Twiddler (Figure 1), a mobile one–handed chording keyboard with a keypad similar to a mobile phone. The Twiddler has twelve keys arranged in a grid of three columns and four rows on the front of the device. The device is held with the keypad facing away from the user and each row of keys is operated by one of the user's four fingers. Additionally, the Twiddler has several modifier buttons such as "Alt", "Shift", "Control", etc. on the top operated by the user's thumb. Instead of pressing keys in sequence to produce a character as with traditional keyboards, multiple keys can be pressed simultaneously to generate a chord.

Figure 1: Chord for the letter "j" (R0L0) on the Twiddler

With traditional keyboards, a character is generated when the corresponding button is pressed. This strategy cannot be used for chording since the user may not press all of the keys for the chord at exactly the same time. Instead, the Twiddler generates the keycode once the first button of a chord is released. Just before that point, all of the buttons for the chord have been depressed so the proper keycode can be generated. In Section , we explore the relationship between the timings of pressing the buttons and how they relate to learning to chord.

The default keymap for the Twiddler is shown in Table 1. The chords consist of single button and two button chords which are assigned in an alphabetical order. The four characters in the Fingers column denote what keys to press from each row. 'L' indicates the leftmost button in a row, 'M' the middle and 'R' the right button. A '0' means the corresponding finger is not used in the chord. The chord for 'a' is 'L000' which indicates that the user should press the left button on the top row. To generate 'j' ('R0L0'), the user would press the right key on the top row and the left key on the third row (Figure 1). Note that the designation for left and right is from the user's perspective while holding the keypad facing away. As a result, there is a left–to–right mirror between the table and the figure.

Fingers	Char	Fingers	Char	Fingers	Char
L000	a				
0L00	b	RL00	i	ML00	r
00L0	c	R0L0	j	M0L0	s
000L	d	R00L	k	M00L	t
M000	e				
0M00	f	RM00	l	MM00	u
00M0	g	R0M0	m	M0M0	v
000M	h	R00M	n	M00M	w
R000	Space				
0R00	Delete	RR00	o	MR00	x
00R0	Backspace	R0R0	p	M0R0	y
000R	Enter	R00R	q	M00R	z

Table 1: Keymap for chording on the Twiddler.

For a chord on the Twiddler, each of the fingers may be in one of four states (pressing one of three buttons, or not pressing anything). Leaving out the "chord" in which no buttons are pressed, there are then $4^4 - 1 = 255$ possible chords using the four main fingers. The modifier buttons operated by the thumb allow more chords. HandyKey includes what we have termed multi–character chords (MCCs) in the default keymap: single chords that generate a sequence of several characters. For instance, there are chords for some frequent words and letter combinations such as 'and', 'the', and 'ing'. Users can also define their own MCCs. We present an evaluation and analysis of the effects of MCCs on expert typing rates in Section .

Previous Work

In our previous work[5], we evaluated the relative learning rates of typing with multi–tap versus typing with chording on the Twiddler. We conducted a longitudinal study with ten participants. None of the participants had any experience with typing chords on the Twiddler. However, they had varying levels of practice typing with multi–tap.

The experiment was a 2 x 20 within–subjects factorial design in which we presented the participants with two conditions (multi–tap and chording) during 20 sessions of typing. A session consists of two parts delineated by typing condition and a five minute break in the middle. Each part of the session, which lasts 20 minutes, consists of several blocks of trials. A block contains ten text phrases of approximately 28 characters each which were selected randomly from the set of 500 phrases developed by MacKenzie and Soukoreff [8]. These are phrases specifically designed as representative samples of the English language. The phrases contain only letters and spaces, and we altered the phrases to use only lower case and American English spellings. The software used for our experiments (Figure 10) is designed to prompt the participant with the phrase to be typed and record the response and timings for all of the buttons pressed.

We found the mean entry rates for our ten participants for session one were 8.2 wpm for multi–tap and 4.3 wpm for chording. As sessions continued, the means improved and reached 19.8 wpm for multi–tap and 26.2 wpm for chording by the end of the study (20 sessions). While both conditions showed improvement, the typing rates for the chording condition rapidly surpassed those of multi–tap (Figure 2). After 20 session it is clear that the learning for multi–tap has tapered off. The regression curves indicate that there is minimal improvement with each additional session. Chording, however, is still showing strong signs of learning.

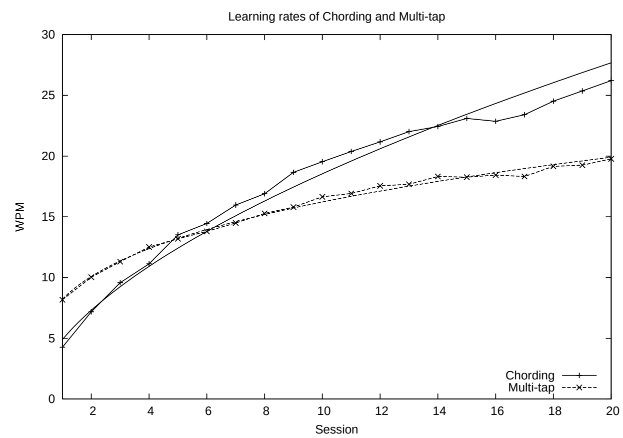

Figure 2: Learning rates and exponential regression curves for multi–tap and chording for 20 sessions [5].

Here we present a follow–up study designed to determine what chording rates our participants could achieve and to confirm or refute the expert rates predicted by our regression curves. We also analyze the nature of how they learned to type with chords. Finally, we examine the use of MCCs by our now "expert" typists and the effects of limited visual feedback.

Learning to Chord

The study presented here continued with a very similar procedure as in our previous work. Five of our original ten participants agreed to continue in the experiment, and we resumed testing after a two week intermission. After continuing with the same procedure for 8 sessions, we replaced the multi–tap condition with a second chording phase. As a result, each time a participant came in we collected two 20 minute sessions of chording data. For this experiment, we compensated each participant at the rate of $0.33 * words per minute * accuracy. In this paper, we only present the data we collected on chording.

Towards Expertise

We collected data for approximately 25 additional 20 minute sessions. This resulted in a total of 45 sessions or about 15 hours of practice per participant for this phase of our experiment. We ended this phase when our participants were showing signs of expertise with reduced rates of learning. Figure 3 shows the average typing speed across participants. Also plotted are the original regression from our first study and a modified regression based on the new data for our five participants.

$$\begin{aligned}\text{Original regression}: y &= 4.8987x^{0.5781}, R^2 = 0.9849\\ \text{Modified regression}: y &= 5.3503x^{0.5280}, R^2 = 0.9787\end{aligned}$$

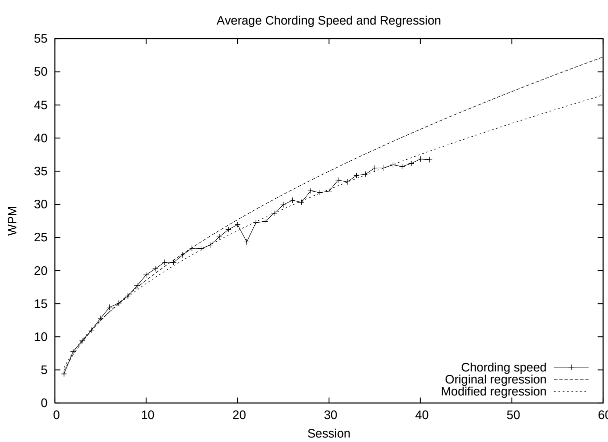

Figure 3: Mean learning rates and regression curves across participants.

The effect of the two week break can be seen as a dip in the typing rate at session 20 but the participants rebound by the next session. This break might account for some of the variation between the curves. Even though there is a good fit to the mean typing rate of the participants, there are large differences in each individual's typing rate. Figure 4 shows the typing speeds and regression curves for each of the participants by session. All of the regressions have correlations of at least 0.96, indicating that the data is well–fitted to the regression curves. They predict that after 60 sessions, even the slowest participants would be able to type at 35 words per minute while the fastest would achieve rates in excess of 65 wpm.

Figure 5 shows the average error rate across participants using Soukoreff's and Mackenzie's total error rate metric [12]. The final mean error rate is 6.2% and is slightly above other typing studies with a similar experimental design [7]. As shown, participants rapidly reduced their error rates as they learned chording. However, as they learned to type faster,

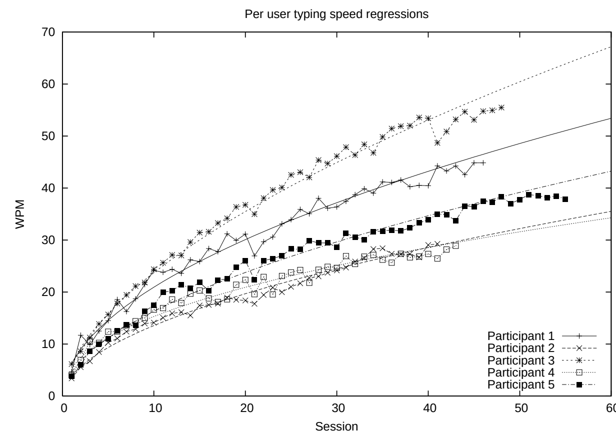

Figure 4: Per user typing rates and regressions

their accuracy gradually decreased. We did not directly control for accuracy. Instead, each participant was compensated proportional to the product of his rate and accuracy. As a result, the participants were rewarded if a small decrease in accuracy enabled a faster typing rate. A similar effect, where errors rates gradually increase as participants became experts, was shown by Matias *et. al* with the Half–QWERTY keyboard [9].

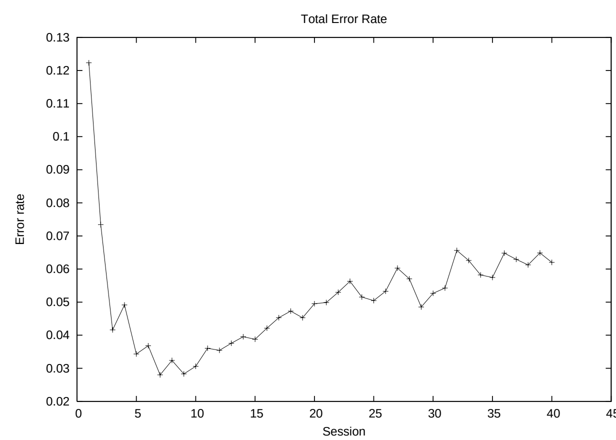

Figure 5: Mean error rate across participants.

Analysis of Learning Rates

Two goals for this phase of experiment were to investigate how users type on the Twiddler and to study the nature of the learning involved with chording. With a traditional keyboard, a character is generated by pressing and releasing a single key. Chord typing, however, may involve pressing and releasing two or more buttons to generate a character. To investigate chords we instrumented our experimental software so that it records the time each button is pressed and released for every chord.

Typing a degenerate chord involving only a single button has one press and one release. This keypress has two intervals associated with it. The first interval is from when the last chord was completed (all of the buttons were released) to when the button for the chord is depressed. This time interval is the "in–air" time as there are no keys being held down. The interval between the press of the button and the release is the "hold" time. We extended this notion to two button chords. The interval during which no buttons are pressed down is the in-air time and the time during which all of the buttons are depressed is the hold time. However, the buttons in the chord may not be pressed or released at the exactly the same moment in time. This introduces two additional intervals. The time between the press of the first and second buttons of the chord is the "press" time interval while the time between releasing the first and second button of a chord is the "release" interval. Thus, the sequence of two button chord time intervals is in–air, press, hold, and release, whereas single buttons only have in–air and hold intervals.

Figure 6 graphs per–session averages of these intervals for a representative participant. This graph highlights where users spend their time in chording and suggests where the improvements of learning have the most effect. These values were computed by taking the intervals for each chord typed in sentences without any errors and then averaged for the whole session on a per user basis. We did not include sentences with errors as we did not want to confound our data on intervals. Mistyping one chord can impact several others, and it is not straightforward to incorporate the error data with our individual time intervals. We intend to examine errors and their effects more thoroughly in future work.

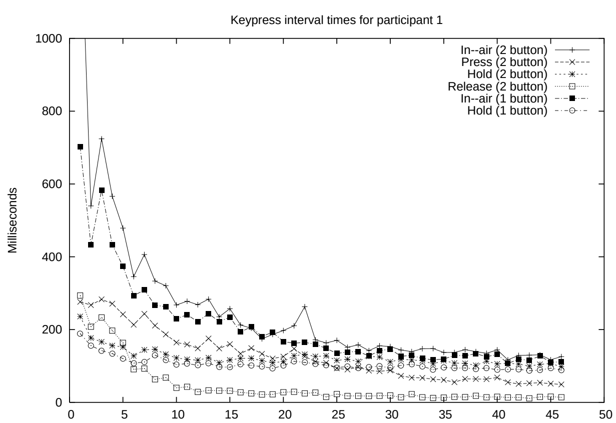

Figure 6: Keypress interval times for a single participant

Figure 7 shows all of the participants' average in–air intervals for single and two button chords. These time intervals exhibit the largest effects of learning. For novices, it is likely that this interval is dominated by the cognitive effort associated with remembering how to type each character and how to move their fingers to the correct position to type the letter. For experts, the delay becomes dominated by the time it takes to move the fingers from one chord to another.

Comparing the in–air interval for single and two button chords reveals that, on a per user basis, the single button times were slightly faster and show better rates of learning. However, the two button in–air interval tracked the single button interval rather well. By the end of the study, the difference between the two times on a per user basis became much smaller which indicates that two button chords do not have a large overhead compared to single button chords for this aspect of chording.

Figure 8 presents the press interval, which is the time between the first and second buttons of a chord being pressed. This interval is particularly interesting because it reveals different typing strategies between users. A single participant (number 3) always pushes both of the buttons in a chord at nearly the exact same time. The average delay between the first and second button press is only 7.25 ms indicating that he always presses both buttons as one action. The other participants show a larger delay between these button presses, indicating that they press the buttons sequentially and likely learned how to press the chords in a different way than participant 3. The delay could be from planning and executing the two button presses in the chord separately. The slower users may also initially wait for feedback from pressing the first button. For these participants there is some learning associated with this interval; however, it is not nearly as pronounced as the in–air time interval learning.

Participant 3 was significantly faster than the other participants and was typing at 67 wpm by the conclusion of our experiments. To see if this might be attributable to his simultaneous press strategy, we examined the data from the other five participants from our original study, who had stopped after 20 sessions. Two of the subjects employed the "simultaneous press" strategy, two of them the sequential strategy, and one started out sequential but appeared to switch mostly over to the simultaneous strategy by the end of the twenty sessions. The participants who used the simultaneous press strategy were no faster than those who used the sequential strategy. However, this is not conclusive evidence that this technique does not help expert typing rates; instead, it indicates that simultaneous pressing might not produce the fastest rates while learning. Once the effects of learning are greatly reduced and the users become experts, the time savings associated by pressing both buttons simultaneously should help improve rates. At 60 words per minute, the average time to type one character is 200 ms. Since the press interval times varied up to 100 ms between users and apply to more than 66% of the alphabet, it appears that there could be significant increase in typing rate by pressing both buttons of a chord at the same time.

Figure 9 shows average hold interval times for single and two–button chords. This interval shows slight improvement with practice. It is slightly faster for single button chords; perhaps participants spend the extra time to ensure that avoid releasing the first finger before the second one is depressed.

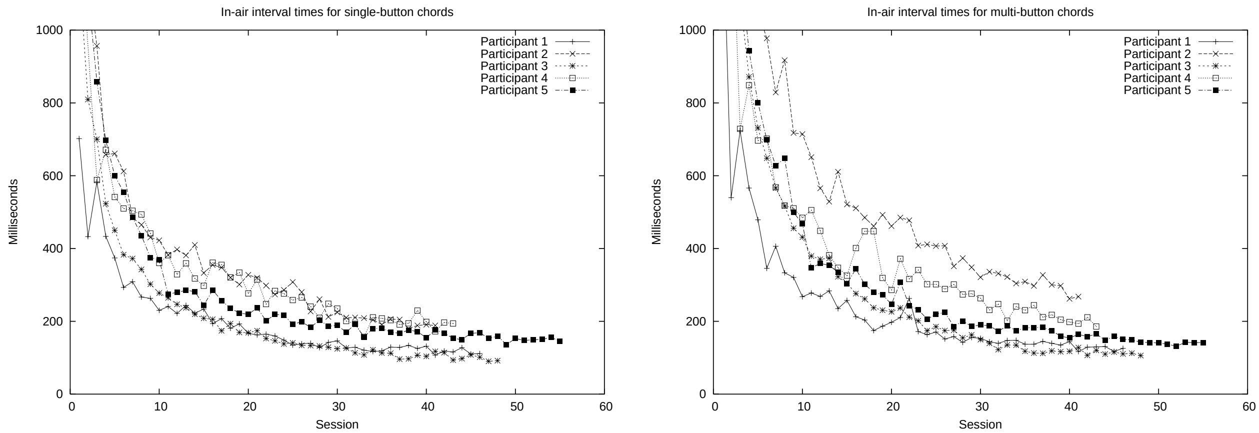

Figure 7: In–air interval times for single button chords (left) and two button chords (right).

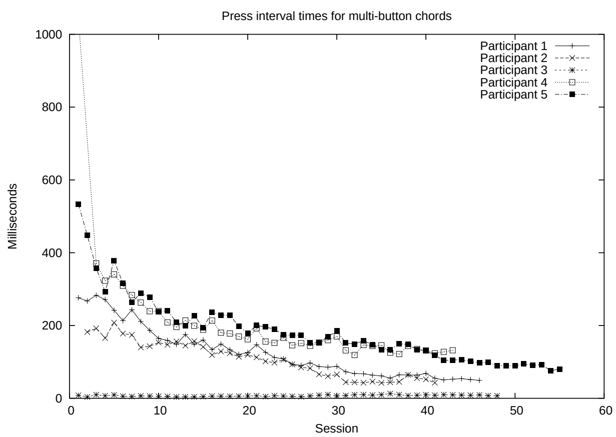

Figure 8: Press interval times (two–button chords)

Our last interval is the release time. While only one participant pressed both keys of a chord at the same time all of the participants rapidly learned to release both buttons of a chord at approximately the same time. After about 10 sessions most of the users were averaging under 25 ms for this interval.

Expert Usage

After about 45 sessions, enough data had been collected that we could be confident of our regressions' predictions. While performance was still improving, the rate of learning had decreased enough that we considered our participants to be expert users. At this point we continued our experiment with two additional phases designed to investigated various aspects of expert typing. We studied the possible benefits of multi–character chords (MCCs). Finally, we examined the effects of typing with reduced visual feedback ("blind typing"). At the conclusion of our experiments, participants completed an average of 75 sessions (25 total hours of practice). The average final typing rate is 47 wpm while the fastest typist achieved 67 wpm.

Multi–Character Chords

There are 255 possible chords that can be typed on the Twiddler using the four fingers. Of these, only a small subset are allocated to the alphabet and punctuation needed to type English text. Some of the unused chords can be employed as Multi–Character Chords (MCCs). These chords could generate any sequence of characters. In the next phase of our experiment we wanted to determine if MCCs for short common words and suffixes would improve our now expert participants' typing rates. Our hypothesis is that MCCs would have a positive impact on typing rate because the number of button presses needed to type any given MCC string, such as "the ", would be reduced down to one chord. This would reduce the overall number of keystrokes per character (KSPC) [6] as fewer keystrokes (button presses) would be needed to generate the same text.

We chose to investigate the benefits of MCCs by selecting 12 common strings. Using word frequency data from the British National Corpus [2] we selected 12 strings of at least three letters that are very prevalent in written English. For this experiment we selected the strings "for", "and", "the", "ent", "ing", "tion", "ter", "was", "that", "his", "all", and "you" to be typed as MCCs. We assigned these strings to unused chords that did not involve the index finger. As many of these strings are normally followed by a space character, this assignment enabled us to add 12 extra MCCs that had a trailing space. The buttons used for these chords are the same as the normal version, only the user also depresses the button used for space (the right button operated by the index finger).

To introduce MCCs to our participants, we modified the experimental software to highlight the next MCC that could be typed. Our software also has a diagram of the Twiddler keypad that was designed to act as a guide to help the users learn the basic alphabet keymap. We modified the diagram so that the keys needed for the MCC are also highlighted (Figure

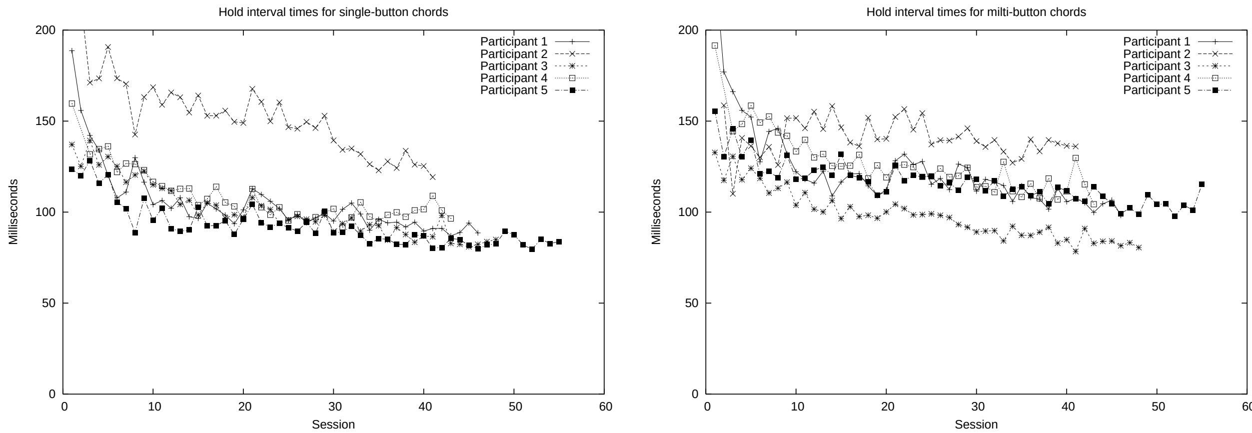

Figure 9: Hold intervals for single button chords (left) and two button chords (right).

10). To encourage the use of MCCs, we modified the error calculation so that typing the MCC string letter–by–letter counted against the participant's accuracy.

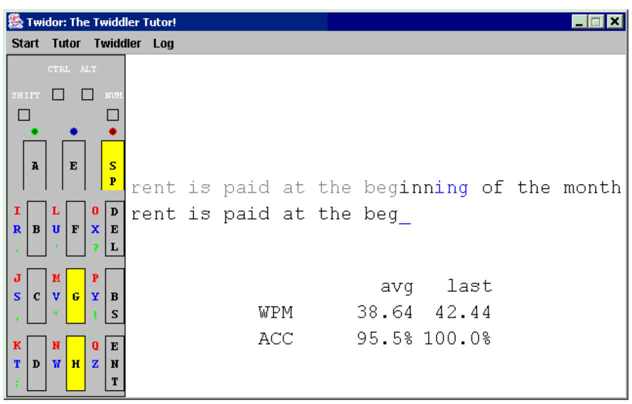

Figure 10: Our experimental software showing the use of MCCs; "ing " is the MCC to be typed ("R0MM") and is highlighted in blue.

We did not find strong evidence that MCCs improve typing rates. Initially, our participants typed more slowly when using MCCs as they were novices for those chords. For the first session, the average typing speed dropped to 83.5% of what it had been. However, on the fifth session, the average speed was 97.1% of the pre–MCC speed, and by the tenth session it was 104.5%, and continued to improve. Even though the rate increased beyond the typing speed just before the introduction of MCCs, the participants were still slowly learning. As a result we cannot attribute the overall increase in rate solely to the effects of MCCs.

To better understand the effects of MCCs we compared the amount of time participants needed to type the MCC strings letter–by–letter just before the introduction of MCCs and the time needed to type the new chord. On average, users type the MCC strings using the multi–character chord in 58.5% of the time it would take to type those same characters letter–by–letter. An analysis of our phrase set revealed that 17.5% of the characters in our phrase set can be typed with MCCs. As a result, weighted by the frequency of MCCs in our phrase set, this would correspond to about an 8% increase in average overall typing speed. This effect would likely get more pronounced as our participants master the new multi–character chords. At the end of the MCC phase of our experiment, our participants were taking an average of 596ms to type each multi–character chord and were still showing signs of improvement with MCCs. While our multi–character chords might be slower in general because they involve multiple buttons, the chords for the alphabet that require two buttons only take 354 ms on average which is only 31.3% more time than typing a single button chord. As a result, we expect MCC rates would improve once our participants mastered the multi–character chords.

Blind Typing

In a mobile environment, a user's visual attention may be diverted away from her display while entering text. Silfverberg examined the effect of visual and tactile feedback when using a mobile phone keypad [11]. He found that with high tactile feedback, finding and pressing a key with limited visual feedback has minimal effects on the average error rate.

Inspired by these results and our own anecdotal experience of typing with limited visual feedback, we designed the last phase of our chording experiment to evaluate blind typing on the Twiddler. We have a 3 x 5 design with 3 conditions (normal feedback, "dots" feedback, and "blind") over 5 sessions of typing. Each condition lasts 15 minutes. Our normal feedback condition displays the text typed under the phrase presented to the participant as in Figure 10, but without MCC highlighting. As the Twiddler is held with the keypad facing away from the user, this condition corresponds most closely to Silfverberg's indirect visual feedback condition. For our "dots" condition, we display periods for each character typed instead of the transcribed text. Thus, participants see their position in the supplied phrase, but not specifically what they type. This condition is designed to simulate monitoring text typed without being able to actually read the letters. Finally, the "blind" condition does not show any on-screen indication of what was being typed and mimics Silfverberg's no visual feedback condition. For both the "dots" and "blind" conditions, participants are shown their transcribed text and error statistics when they press enter at the end of the phrase. We predicted that like Silfverberg, reducing the visual feedback would limit our participants' typing rate and accuracy.

Typing Rates (wpm)					
Participant	1	2	3	4	5
Normal	51.8	37.6	64.2	36.2	41.8
Half–blind	51.7	37.5	**67.2**	36.0	**43.1**
Blind	**53.7**	37.5	**67.7**	36.6	41.7

Table 2: Per participant typing rates for the three conditions. Bold indicates a statistically significant difference at the 0.05 level between that condition and the normal condition for that user.

Percent Errors					
Participant	1	2	3	4	5
Normal	5.61	5.62	7.01	9.83	6.64
Dots	**4.82**	5.02	**5.75**	9.26	5.83
Blind	5.03	**4.63**	**5.90**	8.89	**5.44**

Table 3: Per participant errors for the three conditions. Bold indicates a statistically significant difference at the 0.05 level between that condition and the normal condition for that user.

Surprisingly, changing the visual feedback did not hinder the participants in their typing as expected. In some cases typing rates and error improved with the reduced feedback. Table 2 shows the change in speed and Table 3 the error rate for the blind typing conditions. Values where a two–tailed t–test showed a statistically significant difference at the 0.05 level from the normal condition are marked with bold. Whenever there is a statistically significant difference between normal typing and one of the reduced feedback conditions, the reduced feedback condition shows an improved typing rate or a reduced error rate. More work will be needed to explore which factors effected this result.

Future Work

In the future, we would like to create a model of Twiddler chording which accounts for finger motion and effects between chords. Ideally, we could use this model to evaluate different keymaps and optimize them for various tasks. We wish to continue our study of multiple character chords to determine their effect on overall learning and typing speed. Finally, we would like to analyze the types of errors made while chording in order to create a more effective teaching system.

Conclusion

We have analyzed various aspects of expert chording on the Twiddler keyboard including text entry speed, the effects of visual feedback, and the use of multiple character chords. Given the expert users' high text entry speeds and ability to touch type, chording seems a viable mechanism to explore for text entry on future mobile devices.

Acknowledgements

This work is funded in part by NSF Career Grant #0093291 and the NIDRR Wireless RERC. We would like to thank Scott MacKenzie and William Soukoreff for offering their phrase set for use. We would also like to thank Shumin Zhai and Miika Silfverberg for their comments.

REFERENCES

1. CellularOnline. Women embracing SMS - study. http://www.mobileoffice.co.za/news_2002/100202-women_embracing_sms.htm, October 2002.

2. G. Leech, P. Rayson, and A. Wilson. *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Longman, 2001.

3. M. Lindstom. Message madness our big chance. SMH http://www.smh.com.au, February 2002.

4. K. Lyons. Everyday wearable computer use: A case study of an expert user. In *Proceedings of Mobile HCI*, pages 61–75, 2003.

5. K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. Looney. Twiddler typing: One-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, 2004.

6. I. S. MacKenzie. KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of Mobile HCI 2002*, pages 195–210, 2002.

7. I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner. Letterwise: prefix-based disambiguation for mobile text input. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 111–120. ACM Press, 2001.

8. I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 extended abstracts*, pages 754–755. ACM Press, 2003.

9. E. Matias, I. S. MacKenzie, and W. Buxton. One-handed touch typing on a qwerty keyboard. *Human-Computer Interaction*, 1996.

10. Mobile CommerceNet http://www.mobile.seitti.com, January 2002.

11. M. Silfverberg. Using mobile keypads with limited visual feedback: Implications to handheld and wearable devices. In *Proceedings of Mobile HCI 2003*, pages 76–90, 2003.

12. R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the conference on Human factors in computing systems*, pages 113–120. ACM Press, 2003.

13. T. Starner. *Wearable Computing and Context Awareness*. PhD thesis, MIT Media Laboratory, Cambridge, MA, May 1999.